

sub 017
1. (Three times Amended) A method of searching for a string in a lexical cache, comprising the computer-implemented steps of:

generating a key based on the string;

identifying a lexical container from among a plurality of lexical containers based on a length

of the key, said lexical containers associated with respective key lengths and configured

to hold respective maximum numbers of entries based on the respective key lengths; and

searching the lexical container for an entry associated with the string based on the key,

wherein at least one of the lexical containers is configured to hold a different maximum

number of entries than at least another one of the lexical containers.

2. (Not Amended) The method of claim 1, wherein the step of generating a key based on the string includes the step of compressing the string to produce the key.

3. (Not Amended) The method of claim 2, wherein the step of compressing the string to produce the key includes the step of performing an n-gram compression on the string.

4. (Not Amended) The method of claim 1, wherein the step of generating a key based on the string includes the step of using the string as the key.

5. (Previously Amended) The method of claim 1, wherein the step of identifying a lexical container includes the steps of:

generating a prefix based on the key; and

identifying the lexical container from among the plurality of the lexical containers based on

the length of the key and the prefix.

6. (Not Amended) The method of claim 1, wherein:

the step of identifying a lexical container based on a length of the key includes the step of identifying a hash table based on the length of the key, said hash table containing sequences of slots for holding entries associated with strings, each of said sequences of slots corresponding to a respective hash value; and

the step of searching the lexical container for an entry associated with said string includes the steps of:

computing a hash value based on the key; and

searching the hash table based on the hash value for a slot holding an entry associated with said string.

7. (Not Amended) The method of claim 6, wherein the step of computing a hash value based on the key includes the step of computing the hash value based on the key and a prime number associated with the hash table.

8. (Not Amended) The method of claim 7, wherein the step of searching the hash table based on the hash value includes the steps of:

indexing one or more fixed regions of the hash table, each of the fixed regions having the prime number of slots, based on the hash value to identify one or more respective slots; and

inspecting the one or more respective slots for a respective key value matching the key.

9. (Not Amended) The method of claim 8, wherein the step of searching the hash table further includes the step of searching for the key in a linked list of slots stored in an expansion region of the hash table, if the key was not found in the one or more respective slots for the key.

10. (Not Amended) The method of claim 6, further including the step of, if an entry for the string is not found at a first slot that corresponds to the hash value, but is found in a slot that belongs to a sequence of slots that correspond to keys that produce said hash value, then moving a relative position of the entry for the string within the sequence of slots toward the beginning of the sequence of slots.

11. (Not Amended) The method of claim 6, further comprising the step of initializing a descriptor for the hash table, said descriptor storing a reference to the hash table and parameters for the hash table;

wherein the step of identifying a hash table includes the step of identifying a descriptor indicating the hash table and a prime number.

12. (Not Amended) The method of claim 11, wherein the step of initializing a descriptor for the hash table includes the step of initializing a prime number for use in computing a hash value.

13. (Not Amended) The method of claim 11, wherein the step of initializing a descriptor for the hash table includes the step of initializing a maximum number of slots for the hash table.

14. (Not Amended) The method of claim 11, wherein the step of initializing a descriptor for the hash table includes the step of initializing a maximum length of the sequences of slots for the hash table.

15. (Previously Amended) A method of searching for a string in a lexical cache, comprising the computer-implemented steps of:

compressing the string to generate a key;

identifying a hash table from among a plurality of hash tables based on a length of the key,

said hash table containing sequences of slots for holding respective key values, each of

said sequences of slots corresponding to a respective hash value and a number of slots

being based on a respective key length, wherein at least one of the hash tables is

configured to hold a different number of slots than at least another one of the hash tables;

C¹ computing a hash value based on the key;

using said hash value to locate a beginning of the particular sequence of slots that correspond to said hash value;

searching the particular sequence of slots for a slot holding a key value matching the key; and

if a slot having a key value matching the key is found in the particular sequence of slots, but is not at the beginning of said particular sequence of slots, then moving a relative position of the key value within the particular sequence of slots toward the beginning of the particular sequence of slots.

16. (Three times Amended) A computer-readable medium bearing instructions for searching for a string in a lexical cache, said instructions arranged, when executed by one or more processors, to cause the one or more processors to perform the steps of:

generating a key based on the string;

identifying a lexical container from among a plurality of lexical containers based on a length

of the key, said lexical containers associated with respective key lengths and configured

to hold respective maximum numbers of entries based on the respective key lengths; and

searching the lexical container for an entry associated with the string based on the key,
wherein at least one of the lexical containers is configured to hold a different maximum
number of entries than at least another one of the lexical containers.

17. (Not Amended) The computer-readable medium of claim 16, wherein the step of
generating a key based on the string includes the step of compressing the string to produce the
key.

18. (Not Amended) The computer-readable medium of claim 17, wherein the step of
compressing the string to produce the key includes the step of performing an n-gram compression
on the string.

19. (Not Amended) The computer-readable medium of claim 16, wherein the step of
generating a key based on the string includes the step of using the string as the key.

20. (Previously Amended) The computer-readable medium of claim 16, wherein the step of
identifying a lexical container includes the steps of:

generating a prefix based on the key; and

identifying the lexical container from among the plurality of the lexical containers based on
the length of the key and the prefix.

21. (Not Amended) The computer-readable medium of claim 16, wherein:

the step of identifying a lexical container based on a length of the key includes the step of
identifying a hash table based on the length of the key, said hash table containing

sequences of slots for holding entries associated with strings, each of said sequences of slots corresponding to a respective hash value; and

the step of searching the lexical container for an entry associated with said string includes the steps of:

computing a hash value based on the key; and

searching the hash table based on the hash value for a slot holding an entry associated with said string.

22. (Not Amended) The computer-readable medium of claim 21, wherein the step of computing a hash value based on the key includes the step of computing the hash value based on the key and a prime number associated with the hash table.

23. (Not Amended) The computer-readable medium of claim 22, wherein the step of searching the hash table based on the hash value includes the steps of:

indexing one or more fixed regions of the hash table, each of the fixed regions having the prime number of slots, based on the hash value to identify one or more respective slots;

and

inspecting the one or more respective slots for a respective key value matching the key.

24. (Not Amended) The computer-readable medium of claim 23, wherein the step of searching the hash table further includes the step of searching for the key in a linked list of slots stored in an expansion region of the hash table, if the key was not found in the one or more respective slots for the key.

25. (Not Amended) The computer-readable medium of claim 21, wherein said instructions are further arranged to cause the one or more processors to perform the step of, if an entry for the string is not found at a first slot that corresponds to the hash value, but is found in a slot that belongs to a sequence of slots that correspond to keys that produce said hash value, then moving a relative position of the entry for the string within the sequence of slots toward the beginning of the sequence of slots.

26. (Not Amended) The computer-readable medium of claim 21, wherein said instructions are further arranged to cause the one or more processors to perform the step of initializing a descriptor for the hash table, said descriptor storing a reference to the hash table and parameters for the hash table;

wherein the step of identifying a hash table includes the step of identifying a descriptor indicating the hash table and a prime number.

27. (Not Amended) The computer-readable medium of claim 26, wherein the step of initializing a descriptor for the hash table includes the step of initializing a prime number for use in computing a hash value.

28. (Not Amended) The computer-readable medium of claim 26, wherein the step of initializing a descriptor for the hash table includes the step of initializing a maximum number of slots for the hash table.

29. (Not Amended) The computer-readable medium of claim 26, wherein the step of initializing a descriptor for the hash table includes the step of initializing a maximum length of the sequences of slots for the hash table.

30. (Previously Amended) A computer-readable medium bearing instructions for searching for a string in a lexical cache, said instructions arranged, when executed by one or more processors, to cause the one or more processors to perform the steps of:

compressing the string to generate a key;

identifying a hash table from among a plurality of hash tables based on a length of the key,

said hash table containing sequences of slots for holding respective key values, each of

said sequences of slots corresponding to a respective hash value and a number of slots

being based on a respective key length, wherein at least one of the hash tables is

configured to hold a different number of slots than at least another one of the hash tables;

computing a hash value based on the key;

using said hash value to locate a beginning of the particular sequence of slots that correspond

to said hash value;

searching the particular sequence of slots for a slot holding a key value matching the key; and

if a slot having a key value matching the key is found in the particular sequence of slots, but

is not at the beginning of said particular sequence of slots, then moving a relative position

of the key value within the particular sequence of slots toward the beginning of the

particular sequence of slots.

31. (Twice Amended) A method of storing a string in a lexical cache, comprising the computer-implemented steps of:

generating a key based on the string;

identifying a lexical container from among a plurality of lexical containers based on a length

of the key, said lexical containers are associated with respective key lengths and

configured to hold respective maximum numbers of entries based on the respective key lengths; and
storing the string in an entry in the lexical container based on the key,
wherein at least one of the lexical containers is configured to hold a different maximum number of entries than at least another one of the lexical containers.

C1

32. (Twice Amended) A computer-readable medium bearing instructions for storing a string in a lexical cache, said instructions arranged, when executed by one or more processors, to cause the one or more processors to perform the steps of:

generating a key based on the string;

identifying a lexical container from among a plurality of lexical containers based on a length of the key, wherein the lexical containers are associated with respective key lengths and configured to hold respective maximum numbers of entries based on the respective key lengths; and

storing the string in an entry in the lexical container based on the key, wherein at least one of the lexical containers is configured to hold a different maximum number of entries than at least another one of the lexical containers.

33. (Not Amended) The method of claim 1, wherein:

a first lexical container of the lexical containers is associated with a first key length;

a second lexical container of the lexical containers is associated with a second key length;

the first key length is less than the second key length; and

the first lexical container is configured to hold more entries than the second lexical container.

34. (Not Amended) The method of claim 31, wherein:

a first lexical container of the lexical containers is associated with a first key length;

a second lexical container of the lexical containers is associated with a second key length;

the first key length is less than the second key length; and

the first lexical container is configured to hold more entries than the second lexical container.

35. (Once Amended) A method of providing a lexical cache, comprising the computer-implemented steps of:

/ allocating a plurality of lexical containers each configured to a respective maximum number of entries based on a key length; and

searching for one of the entries associated with a string within one of the plurality of lexical containers corresponding to a key generated based on the string, (')

wherein at least one of the lexical containers is configured to hold a different maximum number of the entries than at least another one of the lexical containers.

36. (Not Amended) The method of claim 35, wherein the maximum number of entries in the allocating step is based on a logarithmic function of the key length.